*aerospace*

MDPI

*Article*

# Research on the Remaining Useful Life Prediction Algorithm for Aero-Engines Based on Transformer–KAN–BiLSTM

**Kejie Xu *, Yingqing Guo and Qifan Zhou**

School of Power and Energy, Northwestern Polytechnical University, Xi'an 710129, China; yqguo@nwpu.edu.cn (Y.G.); george13@mail.nwpu.edu.cn (Q.Z.)
* Correspondence: xukejie@mail.nwpu.edu.cn

**Abstract**

Predicting the remaining useful life (RUL) of aircraft engines is crucial for ensuring flight safety, optimizing maintenance, and reducing operational costs. This paper introduces a novel hybrid deep learning model, Transformer–KAN–BiLSTM, for aero-engine RUL prediction. The model is designed to leverage the complementary strengths of its components: the Transformer architecture effectively captures long-range temporal dependencies in sensor data, the emerging Kolmogorov–Arnold Network (KAN) provides superior approximation flexibility and a unique degree of interpretability through its spline-based activation functions, and the Bidirectional LSTM (BiLSTM) extracts nuanced local temporal patterns. Evaluated on the benchmark NASA C-MAPSS dataset, the proposed fusion framework demonstrates exceptional performance, achieving remarkably low Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) values that significantly surpass existing benchmarks. These results validate the model's robustness and its high potential for practical deployment in prognostics and health management systems.

**Keywords:** remaining useful life prediction; Transformer–KAN–BiLSTM algorithms; aero-engine; multimodal algorithm fusion; advanced predictive algorithms

check for updates

Academic Editor: Wim J. C. Verhagen

Received: 2 September 2025
Revised: 15 October 2025
Accepted: 29 October 2025
Published: 8 November 2025

**Citation:** Xu, K.; Guo, Y.; Zhou, Q. Research on the Remaining Useful Life Prediction Algorithm for Aero-Engines Based on Transformer–KAN–BiLSTM. *Aerospace* **2025**, *12*, 998. https://doi.org/10.3390/aerospace12110998

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

An aero-engine is the core component of modern aircraft, and its performance and reliability directly affect flight safety and operational efficiency [1,2]. With the development of aviation technology, the design and manufacturing levels of aero-engines have been improved continuously, but in the process of long-term use, the performance of the engine will be degraded due to a variety of factors, and the assessment of the remaining useful life is particularly important [3,4]. Remaining useful life (RUL) refers to the period or cycle that an aero-engine is expected to be able to operate safely and efficiently under its current operating conditions. The accurate assessment of RUL is crucial for airlines, maintenance organizations, and manufacturers, as it involves many aspects, such as safety, economy, and reliability.

The concept of advanced prediction and assessment of the remaining useful life of an aero-engine is an extremely important reference criterion for the assessment of the performance of an aero-engine. The effective analysis of the endurance time is an important measure to determine whether the engine can maintain a reliable operation and long-term mode of operation [5], and its importance mainly focuses on the following aspects [6–8]: (1) Safety. Aero-engine failure may lead to catastrophic and disastrous consequences. Therefore, an accurate assessment of its remaining useful life can effectively

prevent accidents from occurring. By monitoring and analyzing the health status of the engine, potential problems can be uncovered in time so that appropriate maintenance measures can be taken. (2) Economy. For airlines, the maintenance and replacement costs of engines are an important part of operating expenses. By reasonably assessing the remaining useful life, airlines can optimize the maintenance plan, reduce unnecessary operating costs, and improve economic efficiency. (3) Reliability. The reliability of aviation engines directly affects the normal operation of flights. Evaluating the remaining useful life can ensure that the engine maintains good performance within the scheduled service life and improves overall reliability.

The research method for aero-engine remaining useful life prediction, in terms of the development concept of the technology, mainly includes the following parts. Firstly, there is the a priori experience-based approach [9–11], which is the earliest method of RUL assessment and mainly relies on historical data and expert experience. The empirical model is established by analyzing the failure data of similar types of engines. The advantage of the technical realization is that it is simple, easy to implement, and applicable to cases with fewer data, but it lacks systematicity and accuracy. The realization is mainly achieved through the statistical analysis of past operating data, establishing the relationship model between the engine life and the conditions of use, combined with the knowledge and experience of experts, to assess the RUL under different operating conditions. The second approach is based on physical models [12–14], where the physical modeling of the physical characteristics of the engine and the working mechanism is conducted to analyze its performance degradation law. Firstly, based on the thermodynamic principle, we analyze the performance change in the engine under different working conditions, and through the material fatigue theory, we analyze the wear and damage of the engine components in long-term operation. The final evolution is based on the data-driven approach, which has become the mainstream RUL evaluation in recent years, utilizing big data and artificial intelligence technology to analyze a large amount of operating data. It is mainly divided into supervised learning, which uses labeled data to train models and predicts RUL through regression analysis and classification models, and unsupervised learning, which uses clustering and other techniques to discover potential patterns in the data for health status monitoring.

In recent years, numerous studies have explored machine learning and deep learning techniques for aero-engine RUL prediction. While these approaches have improved prediction accuracy, they continue to face limitations in terms of generalization, data heterogeneity, and interpretability. To better situate our work within the existing research, Table 1 summarizes representative studies, highlighting the algorithms used, datasets, advantages, and unresolved issues. This comparison clearly shows that most existing models either rely on fixed activation functions with limited adaptability or lack transparent interpretability, motivating our development of the Transformer–KAN–BiLSTM fusion framework, which integrates long-range dependency modeling, adaptive nonlinear mapping, and bidirectional feature extraction.

Wang et al. [15] introduced an integrated deep learning model for aircraft engine RUL prediction, which combines a one-dimensional convolutional neural network (CNN) with a bidirectional long short-term memory network incorporating an attention mechanism (Bi-LSTM-AM). They further employed Bayesian optimization to tune the hyperparameters of the model, thereby enhancing its predictive performance. Meanwhile, Xu et al. [16] developed a novel lightweight multiscale broad learning system (MSBLS), in which elastic-net regularization was applied to constrain the output weights of nodes, preserve effective nodes, and ultimately produce a sparser model. In a different approach, Guo et al. [17] proposed a multiscale Hourglass-Transformer framework for RUL prediction. This architecture uses a one-dimensional CNN to rescale time series into multiple temporal resolutions

for feature fusion. The Hourglass-Transformer then performs further feature extraction from the fused representation to estimate the RUL. With the development of data science, machine learning, and artificial intelligence technologies, more and more predictive algorithms have been applied to aero-engine RUL prediction. However, although these algorithms have improved the accuracy and reliability of prediction to some extent, they still have several limitations and shortcomings.

**Table 1.** Summary of representative studies on aero-engine RUL prediction.

| Study | Method/Model | Key Contribution | Limitation/Research Gap |
| --- | --- | --- | --- |
| Li et al. [6] | CNN-based RUL estimation | Automatically extracts temporal–spatial features from raw signals | Limited interpretability; prone to overfitting |
| Ellefsen et al. [4] | Semi-supervised deep architecture | Combines labeled and unlabeled data for RUL estimation | Weak domain adaptation across engine types |
| Deng et al. [5] | LSTM with long–short feature processing | Captures temporal dependencies effectively | Sensitive to heterogeneous sensor distributions |
| Wang et al. [15] | CNN–BiLSTM with attention mechanism | Enhances multiscale temporal fusion | Attention weights lack physical interpretability |
| Guo et al. [17] | Hourglass-Transformer | Multiscale feature extraction using attention fusion | High computational cost; limited adaptability |
| Xu et al. [16] | Multiscale Broad Learning System | Lightweight model with elastic-net regularization | Restricted nonlinear fitting capacity |

Sharma et al. [18] proposed a framework based on a machine learning approach in an I4.0 environment for predicting the remaining useful life of an aero-engine. Six machine learning models were applied to a dataset containing four subsets of the C-MAPSS dataset, FD001, FD002, FD003, and FD004, and each model was applied to a different degradation condition of the turbofan engine. For FD001, the Random Forest model achieved the lowest RMSE (11.59), and for FD002, FD003, and FD004, the LGBM classifier achieved the lowest RMSE (12.78, 7.95, and 11.04), respectively; Arunan et al. [19] proposed a new model based on temporal dynamics learning for detecting change points in individual devices even under variable operating conditions, utilizing the learned change points to improve the accuracy of RUL estimation. During offline model development, multivariate sensor data were decomposed to learn fused time-dependent features that are generalizable to represent normal operating dynamics under multiple operating conditions; Furqon et al. [20] proposed a multi-domain adaptation (MDAN) framework. MDAN consists of a three-stage mechanism in which a hybrid strategy is used not only to regularize the source and target domains but also to build the intermediate hybrid domains where the source and target domains are aligned. The self-supervised learning strategy is implemented to prevent supervised model collapse. MDAN was critically evaluated by comparing it with recently published work on dynamic RUL prediction. Maulana et al. [21] computed the engine health, estimated the engine's end-of-life (EoL), and finally predicted its remaining useful life (RUL). The proposed algorithm uses a mixture of metrics for

feature selection, logistic regression for health index estimation, and unscented Kalman filtering (UKF) for updating the prognostic model to predict RUL recursively.

(1) Firstly, there is the problem of data heterogeneity. Researchers use the C-MAPSS dataset disclosed by NASA, which has the drawback of data heterogeneity. In practice, the operation data of aero-engines may come from different types of sensors, different test environments, and different operating conditions. This data heterogeneity creates challenges for models when processing such diverse data.

(2) Overfitting due to irrational algorithm fusion: most of the studies used a machine learning foundation and deep learning framework fusion in a random combination, did not consider the connectivity between the underlying architecture and the ability to mine the feature information of the data itself. As a result, the model is prone to overfitting phenomenon due to the model learning the noise and chance in the training data rather than the real underlying law. Overfitting leads to a decrease in the stability and credibility of predictions.

(3) Weak generalization ability: RUL prediction models are usually trained on specific aircraft models or under specific operating conditions. When applied to engines of different models or different operating conditions, the predictive performance of the model decreases substantially. This domain adaptation problem limits the model's ability to generalize.

(4) The algorithmic model lacks interpretability: Most of the models proposed in existing studies are black-box approaches with poor interpretability, making it difficult to understand their design rationale and internal mechanisms.

The multimodal fusion concept of the algorithm combination proposed in this paper featuring KAN, Transformer and BiLSTM algorithms effectively addresses many of the above-mentioned deficiencies and shortcomings.

The Transformer model, known for its ability to capture long-range dependencies through self-attention mechanisms, addresses the challenge of modeling temporal relationships over extended periods. In contrast, KAN introduces a flexible feature-mapping mechanism using B-splines and multi-layer perceptrons (MLPs), allowing the model to adaptively handle complex, heterogeneous input data. The BiLSTM, with its bidirectional architecture, efficiently captures both past and future temporal dependencies, enabling a better prediction of short-term trends in engine degradation. By combining these three models, we achieve a synergistic framework that leverages the strengths of each component. The Transformer handles long-term dependencies, KAN adapts to varying feature scales and provides interpretability, and the BiLSTM effectively models short-term patterns. This hybrid architecture significantly outperforms existing hybrid models, which often struggle with overfitting, limited generalization, and a lack of interpretability. Specifically, our model excels in handling data heterogeneity, improving prediction accuracy, and offering transparency in the decision-making process.

First of all, the KAN algorithm, which combines the dual advantages of splines and MLPs, adheres to the concept of the internal degrees of freedom of splines in the internal framework of the algorithm [22,23], being accurate in low-dimensional functions, easy to adjust locally, and able to switch between different resolutions. It can optimize the feature mapping obtained from learning to a very high degree of accuracy (internal similarity with splines), i.e., it can approximate univariate functions very well; meanwhile, the external inherits MLPs' advantage of being strong in anti-interference for the dimension of feature information, which not only learns the features (external similarity with MLPs), but also learns the structure of the combination of multiple variables. Moreover, the KAN algorithm has strong interpretability [24], originating from its sparse preprocessing, visualization process, pruning operation, and symbolification operation, which visualizes the black box as the white box, thus elaborating the concept of interpretability.

The Transformer architecture, on the other hand, has more obvious advantages in dealing with long dependencies, parallel computing, scaling, and generalizability [25,26]. Among them, the Transformer's scheme for capturing long-distance dependencies without limiting the length of the sequence has a very strong mapping relationship for the long trend analysis of the remaining engine service life, and the Transformer model can process the input data in parallel, which significantly improves the speed of training. Moreover, the multi-attention mechanism of the Transformer enables the algorithms to be more efficient in terms of generalization and generalization [27]. The multi-head attention mechanism in Transformer improves the algorithmic modeling capability in terms of generalization and scalability and enhances the ability to dynamically capture data variability and contextual information.

The BiLSTM algorithm has a strong advantage in the prediction of shorter trends and algorithmic architecture implementation [28], in which some response characteristics in the aero-engine dataset show a more discrete static distribution, representing short-term trends. Therefore, the BiLSTM is more commonly used, and the design of the bilateral channel enables bidirectional feature extraction from both past to present and future to present [29]. The design of the bilateral channels ensures that the feature-capturing ability is fully utilized.

The structure of this paper is mainly divided into the following areas. The first section is the introduction, which explains the principle and necessity of aero-engine remaining life prediction. At the same time, the research by domestic and foreign scholars is elaborated upon, summarized and acknowledged with regard to its limitations, to lead to the content of this paper. Second, this paper introduces the characteristic parameters used for predicting and analyzing the remaining useful life (RUL) based on the C-MAPSS data provided by NASA. Then, the key technologies of this paper's are described, and the idea structure and technical route are realized. Finally, the results are verified and the evaluation indexes are analyzed.

This paper mainly includes the following sections. First, the feature distribution of the C-MAPSS dataset and the overview of the dataset are described. Secondly, the principle and process of each key technology based on the Transformer + KAN + BiLSTM algorithm are introduced. Thirdly, the design concepts and technical routes of the fusion algorithms studied in this paper are investigated. Finally, the fusion algorithms are utilized with the help of the FD001,FD002, FD003, and FD004. Then, the prediction results of each RUL algorithm are analyzed and compared, and the comparison test is verified using the evaluation indexes RMSE, MAPE, and so on.

## 2. Mechanistic Analysis of RUL Influences and Overview of the C-MAPSS Dataset

### *2.1. RUL Influencing Factors and Mechanisms*

There are many factors affecting the remaining useful life [3,30]. These mainly include the following: (1) material performance, where the strength, toughness, high-temperature resistance, and oxidation resistance of the material directly affect the fatigue life of engine components and corrosion resistance. Commonly used high-temperature alloys, ceramic matrix composites, etc., in a high-temperature environment, may experience performance decline that leads to component failure; (2) the working environment, where frequent startups and stops can lead to thermal cycling fatigue, accelerating the aging of the material and the formation of cracks. At the same time, high-altitude flights can lead to a sudden drop in temperature, resulting in thermal stresses on the material; (3) operating conditions, where oxidation and the thermal fatigue of the material are significantly accelerated at high temperatures, especially in the combustion chamber and turbine area, and under

high-pressure conditions, the material is subjected to increased stress, accelerating the formation and expansion of fatigue cracks.

As for the mechanism analysis, the remaining useful life of an aero-engine is affected by several physical and chemical processes that work together to cause material fatigue and damage. Fatigue is one of the main causes of failure in aero-engine components [13,14]. Materials experience repeated stresses under cyclic loading, which ultimately lead to the formation and expansion of microscopic cracks. Meanwhile, high-temperature oxidation is one of the important mechanisms for aero-engine material failure, especially in the combustion chamber and turbine regions. The oxidation process consumes the material, forms oxide layers, and reduces the material's strength. In the oxidized environment, the coupling of corrosion and fatigue accelerates material damage, leading to early failure.

### *2.2. Overview of the C-MAPSS Dataset*

The C-MAPSS dataset is simulated. This is due to the complexity of the aero-engine configuration, the complexity and variability of its air paths, and its classified nature. NASA used the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS, Version 2.0) software to generate this dataset, which was designed to test the performance of different models in conjunction with the operational characteristics of the engine [31,32]. The structure of NASA's proposed turbofan engine degradation monitoring dataset (C-MPASS) is shown in Figure 1 [33]. The main components contain the fan, low-pressure compressor (LPC), high-pressure compressor (HPC), combustion chamber, high-pressure turbine (HPT), low-pressure turbine (LPT), and nozzle.

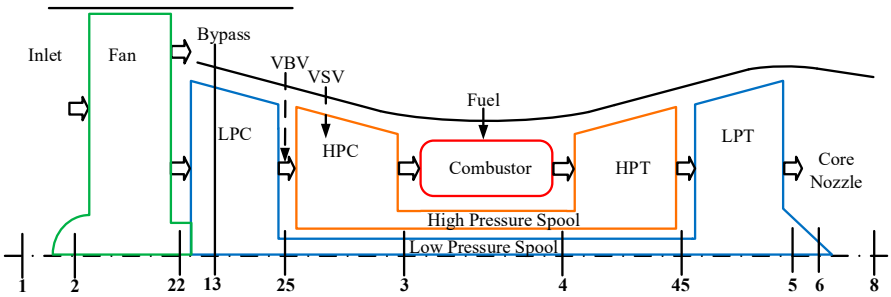


**Figure 1.** Aero-engine systems.

There are four sub-datasets, and each sub-category has a different number of operating conditions and fault states. The C-MAPSS data are shown in Table 2.

**Table 2.** C-MAPSS dataset information.

| Data Set | FD001 | FD002 | FD003 | FD004 |
|---|---|---|---|---|
| Training Set | 100 | 260 | 100 | 249 |
| Testing Set | 100 | 259 | 100 | 248 |
| Working Condition | 1 | 6 | 1 | 6 |
| Fault State | 1 | 1 | 2 | 2 |

Taking FD001 as an example, it is further divided into training and testing subsets, which contain one fault state and one operating condition. The training set Train_FD001.txt contains the parameter information of 100 engines throughout their full life cycles; the test set Test_FD001.txt contains the parameter information of 100 engines in a non-full life cycle state, i.e., it only contains multiple sensor data terminated at a certain time before the engine fails, and the RUL of each engine is predicted in real time based on the given operating parameters. RUL_FD001.txt contains the real values of the RUL of each of the 100 engines

in the test set, which are predicted in real time based on the given operating parameters. The parameter information of each engine contains three operating condition monitoring parameters (flight altitude, Mach number, and throttle stick angle) and 21 performance monitoring parameters, and its 24 sensor monitoring parameters are shown in Table 3.

**Table 3.** Characterization parameters of the C-MAPSS dataset.

| Serial Number | Notation | Specific Interpretation |
|---|---|---|
| 1 | H | Mach number |
| 2 | Ma | Throttle lever angle |
| 3 | TRA | Fan inlet temperature |
| 4 | T2 | Low-pressure compressor outlet temperature |
| 5 | T24 | High-pressure compressor outlet temperature |
| 6 | T30 | Low-pressure turbine outlet temperature |
| 7 | T50 | Fan inlet pressure |
| 8 | P2 | Outer culvert total pressure |
| 9 | P15 | High-pressure compressor outlet gross pressure |
| 10 | P30 | Uncorrected fan speed |
| 11 | NF | Uncorrected core speed |
| 12 | NC | Engine pressure ratio |
| 13 | EPR | High-pressure compressor outlet static pressure |
| 14 | PS30 | Fuel flow and P30 ratio |
| 15 | PHI | Corrected fan speed |
| 16 | NRF | Core engine corrected RPM |
| 17 | NRC | Culvert ratio |
| 18 | BPR | Combustion chamber gas ratio |
| 19 | FARB | Induction gas enthalpy |
| 20 | HT_BLEE | Fan speed command value |
| 21 | NF_DMD | Fan correction RPM command value |
| 22 | PCNFR_DMD | High-pressure turbine cooling air flow |
| 23 | W31 | Low-pressure turbine cooling air flow rate |
| 24 | W32 | Mach number |

The operating data of each engine consist of several parameters, including the following main types of characteristics: Running Time and Life: the cumulative running time (cycles) of the engine and the time it has been in use; Sensor Readings: data involving complex engine operating conditions, such as compressor inlet temperature, turbine outlet pressure, oil temperature, and oil pressure; Engine Speed: RPM (revolutions per minute), showing the operating status of the engine; Air Flow: data on airflow through the engine; Fault Labeling: fault status at each point in time to help identify which data are normal and which are faulty.

These features are organized into a multidimensional matrix in the dataset, allowing each instance of data to be easily analyzed and modeled.

Failure Models:

The C-MAPSS dataset simulates a variety of failure modes which provide rich training samples for failure prediction models. Common types of failures include fan wear, which results in reduced fan efficiency and in turn affects the overall engine performance.

Combustion chamber failures can be caused by fuel supply interruptions or ignition problems, leading to incomplete combustion. Sensor failures, including measurement errors in sensors, can lead to the distortion of critical data, affecting monitoring accuracy. Progressive failures include the wear and tear or aging of components, where a degradation in performance is gradually apparent over time.

## 3. Key Technology and Technical Route of RUL Prediction Algorithm Based on Transformer + KAN + BiLSTM

### *3.1. Predictive Algorithms Key Technology Principles and Concepts*

#### 3.1.1. KAN Algorithm Key Technology

The Kolmogorov–Arnold Network (KAN) is a powerful tool for modeling complex, nonlinear relationships between input features and the predicted output. In the context of RUL ( remaining useful life) prediction, KAN plays a vital role in handling heterogeneous sensor data and capturing intricate patterns of engine degradation. The KAN model consists of two main components [34]: the B-spline layer and the multi-layer perceptron (MLP) layer, which together allow the network to adapt to varying data distributions and learn both short-term and long-term trends in the data.

If $f$ is a multivariate continuous function on a bounded domain, then $f$ can be written as a composite univariate and additive binary operation of a finite number of constant functions. That is, any continuous function $f(x_1, \ldots, x_n)$ can be expressed as a nested combination of a finite number of univariate functions, as shown in Equation (1).

$$f(x) = \sum_{q=1}^{2n+1} \Phi_q \left( \sum_{p=1}^{n} \varphi_{q,p}(x_p) \right) \tag{1}$$

where $\phi_{q,p}$, $\phi_q$ are both univariate functions.

Further, for a smooth $[0,1]^n \to \mathbb{R}$, as shown in Equation (2).

$$f(\mathbf{x}) = f(x_1, \cdots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left( \sum_{p=1}^{n} \phi_{q,p}(x_p) \right) \tag{2}$$

where $x_p$ represents the $q$th element of the vector $x$, so $p$ ranges from 1 to $n$ ($n$ is the dimension of the input vector), and the $q$ index is used to traverse each component of the external function $\phi$. Therefore, there is a unitary function $\phi_{q,p}$ that handles the $q$th component of the input vector $\mathbf{x}$ and contributes a term to the summation of the $q$ external functions.

Also, in the KAN algorithm, there is an important concept of the B-spline function, which is based on the idea of splicing **together** multiple segmented polynomials, each of which is defined by a set of control points (grid points). A set of basis functions is used to represent the spline. These basis functions are locally supported, i.e., each basis function is nonzero on only a few subintervals. In a B-spline, the functions have the same continuity at the nodes (Knot) throughout their domain of definition, and their polynomial expression can have a Cox-de Boor recursive formula expression:

$$\begin{aligned} B_{i,0}(x) &:= \begin{cases} 1 & if\ t_i \le x < t_{i+1} \\ 0 & \text{otherwise.} \end{cases} \\ B_{i,k}(x) &:= \frac{x - t_i}{t_{i+k} - t_i} B_{i,k-1}(x) + \frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(x) \end{aligned} \tag{3}$$

The KAN algorithm, on the other hand, combines the above ideas by introducing the B-spline into the mix and constructing a supervised learning task consisting of input–output

pairs $\{x_i, y_i\}$ requires constructing a $f$ such that for all data points, there is a $y_i \approx f(x_i)$. Determining the appropriate $\phi_{q,p}$ and $\phi_q$ as a means of constructing a machine learning algorithm requires parameterizing Equation (2). Thus, it is necessary to parametrize the expression of each one-dimensional function as a B-spline curve, where the idea and process of substitution is shown in Figure 2.

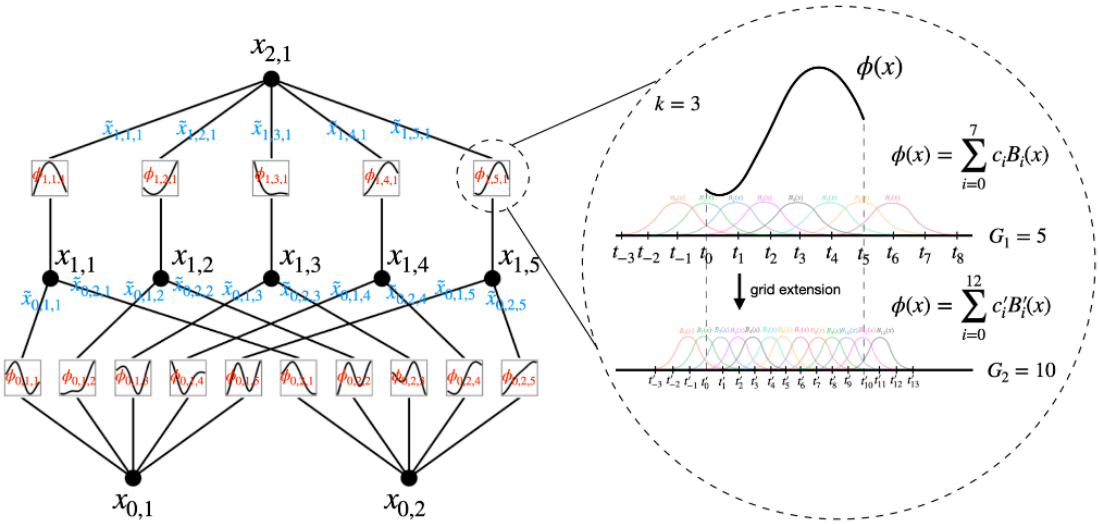


**Figure 2.** B-spline basis function learnable coefficients.

In carrying out the construction process of the deep neural network for the KAN algorithm in detail, it is necessary to generalize the KAN algorithm in a deeper and wider way from two layers of 2n + 1. The KAN layer is defined as a one-dimensional function matrix as shown in Equation (4).

$$\mathbf{\Phi} = \{\phi_{q,p}\}, \quad p = 1, 2, \cdots, n_{\text{in}}, \quad q = 1, 2 \cdots, n_{\text{out}}, \tag{4}$$

where the function $\phi_{q,p}$ has trainable parameters.

In the Kolmogorov–Arnold theorem, there is the idea of "scattering" before "gathering". The internal function $\phi_{q,p}$ forms a KAN layer with input dimension $n_{in} = n$ and output dimension $n_{out} = 2n + 1$, which indicates that each input variable $x_p$ is transformed by a set of functions, and the number of outputs is two times the number of inputs plus one, which is designed to fully capture the information of the input features and transform them into intermediate representations. The external function $\phi_q$ forms a KAN layer with input dimension $n_{in} = 2n + 1$ and output dimension $n_{out} = 1$. This layer is designed to integrate all the outputs of the internal function layer to form the final model output. So far, the combination of two KAN layers is obtained, as shown in Equation (2).

In order for the KAN to reach a sufficient level of depth, for the interpretation on the left-hand side of Figure 2, the shape of the KAN is represented by an array of integers: $[n_0, n_1, \cdots, n_L]$.

Here, $n_i$ is the number of nodes in layer $i$ of the computational graph, and it is denoted by $(l, i)$ for the $i$th neuron in layer $l$. The activation value of the $(l, i)$ neuron is denoted by utilizing $x_{l,i}$. Between layers $(l, i)$ and $(l + 1, i)$, there are $n_l n_{l+1}$ activation functions. The activation function connecting $(l, i)$ and $(l + 1, i)$ is given by the following equation:

$$\phi_{l,j,i}, \quad l = 0, \cdots, L - 1, \quad i = 1, \cdots, n_l, \quad j = 1, \cdots, n_{l+1} \tag{5}$$

The domain activation of $\phi_{l,j,i}$ is $x_{l,i}$, i.e., $x_{1,1}, x_{1,2}, x_{1,3}, x_{1,4}, x_{1,5}$, and the tail activation is $\widetilde{x}_{1,1,1}, \widetilde{x}_{1,2,1}, \widetilde{x}_{1,3,1}, \widetilde{x}_{1,4,1}, \widetilde{x}_{1,5,1}$. The activation value of the $(l+1,i)$th neuron, i.e., $x_{2,1}$, is a composite of all incoming tail activations, as shown in Equation (6).

$$x_{l+1,j} = \sum_{i=1}^{n_l} \widetilde{x}_{l,j,i} = \sum_{i=1}^{n_l} \phi_{l,j,i}(x_{l,i}), j = 1, \cdots, n_{l+1} \tag{6}$$

Equation (7) expresses this composite in matrix form.

$$\mathbf{x}_{l+1} = \underbrace{\begin{pmatrix} \phi_{l,1,1}(\cdot) & \phi_{l,1,2}(\cdot) & \cdots & \phi_{l,1,n_l}(\cdot) \\ \phi_{l,2,1}(\cdot) & \phi_{l,2,2}(\cdot) & \cdots & \phi_{l,2,n_l}(\cdot) \\ \vdots & \vdots & & \vdots \\ \phi_{l,n_{l+1},1}(\cdot) & \phi_{l,n_{l+1},2}(\cdot) & \cdots & \phi_{l,n_{l+1},n_l}(\cdot) \end{pmatrix}}_{\Phi_l} \mathbf{x}_l, \tag{7}$$

where $\phi_l$ is the function matrix corresponding to layer $l$ (B-spline function matrix) and $x$ is the input matrix.

A general KAN network consists of $L$ layers, and given an input vector $\mathbf{x}_0 \in \mathbb{R}^{n_0}$, the KAN outputs are $\mathrm{KAN}(\mathbf{x}) = (\mathbf{\Phi}_{L-1} \circ \mathbf{\Phi}_{L-2} \circ \cdots \circ \mathbf{\Phi}_1 \circ \mathbf{\Phi}_0)\mathbf{x}$. A simplified KAN would then be of the following form: $f(x) = \mathbf{\Phi}_{\text{out}} \circ \Phi_{\text{in}} \circ \mathbf{x}$. Rewrite the above equation similarly to Equation (2), assuming that the output dimension $n$ is 1, and define $f(\mathbf{x}) \equiv \mathrm{KAN}(\mathbf{x})$.

$$f(\mathbf{x}) = \sum_{i_{L-1}=1}^{n_{L-1}} \phi_{L-1,i_L,i_{L-1}} \left( \sum_{i_{L-2}=1}^{n_{L-2}} \cdots \left( \sum_{i_2=1}^{n_2} \phi_{2,i_3,i_2} \left( \sum_{i_1=1}^{n_1} \phi_{1,i_2,i_1} \left( \sum_{i_0=1}^{n_0} \phi_{0,i_1,i_0}\left(x_{i_0}\right) \right) \right) \right) \cdots \right) \tag{8}$$

For the final KAN algorithm to achieve optimal performance and model architecture, the following three-part improvement is needed, starting with a residual activation function that consists of a basis function $b(x)$ (similar to a residual link) such that the activation function $\phi(x)$ is the sum $\phi(x) = w(b(x) + \mathrm{spline}(x))$ of the basis function $\phi(x)$ and the spline function. For the former, $b(x) = \mathrm{silu}(x) = x/(1+e^{-x})$ is set, and for the latter, spline(x) is parameterized as a linear combination of B-splines such that $\mathrm{spline}(x) = \sum_i c_i B_i(x)$, where $c_i$ is trainable. Eventually, $\omega$ can be absorbed into $b(x)$ and $\mathrm{spline}(x)$.

The second step is to initialize the scale, where each activation function is initialized with the value $\mathrm{spline}(x) \approx 0$ and $\omega$ is initialized according to the Xavier initialization. The third step is the update of spline grids, which updates each grid in real time based on its input activation to resolve splines defined over bounded regions.

To illustrate the adaptability of B-splines within the KAN framework, consider an input feature $x$ representing turbine temperature and an output $y$ denoting predicted degradation or RUL. When the underlying relation between $x$ and $y$ is approximately linear, the B-spline basis functions automatically align to produce a near-linear mapping. However, if the degradation behavior changes nonlinearly—such as a rapid performance drop after a certain temperature threshold—the spline introduces additional local curvature around the knot region without altering the global structure. This local adjustability enables the network to capture both smooth and abrupt changes in operational data, improving its flexibility and generalization.

Unlike traditional multi-layer perceptrons that rely on fixed activation functions (e.g., ReLU or tanh), the B-spline formulation provides data-adaptive activation behavior, allowing each neuron to fine-tune its response according to the complexity of the feature.

This mechanism underlies the enhanced adaptability and interpretability of the KAN component [35].

#### 3.1.2. Transformer Algorithm Key Technology

The Transformer components and principles are as follows:

Self-Attention: This is one of the core concepts of the Transformer, which allows the model to consider all positions in the input sequence simultaneously instead of processing them step by step like a recurrent neural network (RNN) or convolutional neural network (CNN). The self-attention mechanism allows the model to assign different attentional weights based on different parts of the input sequence to better capture semantic relationships [36].

Multi-Head Attention: The self-attention mechanism in Transformer is extended with multiple attention heads, each of which can learn different attention weights to better capture different types of relationships. Multi-head attention allows the model to process different information subspaces in parallel.

Stacked Layers: Transformers are usually made up of multiple identical encoder and decoder layers stacked on top of each other. These stacked layers help the model learn complex feature representations and semantics.

Positional Encoding: Since the Transformer does not have built-in sequence position information, it requires additional positional encoding to express the positional order of words in the input sequence.

Residual Connections and Layer Normalization: These techniques help mitigate the problem of gradient vanishing and explosion during training, making the model easier to train.

Encoder and Decoder: Transformers typically include an encoder for processing the input sequence and a decoder for generating the output sequence, which makes them suitable for sequence-to-sequence tasks such as machine translation.

The structure of the Transformer:

When Nx = 6, the encoder block consists of six encoders stacked together. A box in the figure represents the internal structure of an encoder composed of multi-head attention and a fully connected feed-forward neural network. As shown in Figure 3, the encoding component of the Transformer is composed of six encoders stacked on top of each other, and the same is true for the decoder. All the encoders are structurally identical, but they do not share parameters between them.

The second component is the self-attention mechanism, which functions as follows.

Sequence modeling: Self-attention can be used for sequence data modeling. It captures the dependencies at different locations in the sequence to better understand the context.

Parallel computing: Self-attention can be computed in parallel, which means it can be efficiently accelerated on modern hardware. It is easier to train and reason efficiently on hardware such as GPUs and TPUs than sequential models such as RNNs and CNNs.

Long-Distance Dependency Capture: Traditional recurrent neural networks (RNNs) may face the problem of gradient vanishing or gradient explosion when dealing with long sequences. Self-attention can handle long-distance dependencies better as it does not need to process input sequences sequentially. After obtaining the matrix Z through the multiple attention mechanism, it is not directly passed into the fully connected neural network, but it goes through an Add & Normalize step.

The Add & Norm layer consists of two parts, Add and Norm, which are calculated as follows:

$$\text{LayerNorm}(X + \text{MultiHeadAttention}(X))\text{LayerNorm}(X + \text{FeedForward}(X)) \quad (9)$$

where $X$ denotes the input of multi-head attention or the feed-forward network. MultiHead Attention($X$) and FeedForward($X$) denote the outputs (the outputs have the same dimensions as the inputs $X$, so they can be added together). The Add method adds a residual block $X$ on top of z. The purpose of adding the residual block is to prevent degradation in the training process of deep neural networks, which means that the deep neural network gradually decreases the loss by increasing the number of layers of the network. Then, it stabilizes to reach saturation, after which it continues to increase the number of layers of the network, causing the loss to increase instead of decrease.

The ResNet residual neural network is introduced in the residual network, and neural network degradation refers to the fact that after reaching the optimal number of network layers, the neural network continues to be trained, increasing loss. For the redundant layers, it is necessary to ensure that constant mapping is carried out for the extra network. Only after constant mapping can we ensure that the extra neural network will not affect the model. The residual connection is mainly to prevent network degradation.

The next step is the construction of the fully connected layer, which is as follows:

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{10}$$

The fully connected layer is a two-layer neural network that is first linearly transformed, then ReLU nonlinear, and then linearly transformed.

This two-layer network maps the input Z to a higher-dimensional space filtered by the nonlinear function ReLU and then changed back to the original dimension after filtering. After six encoder inputs to the decoder, the final module is the decoder, which, like the encoder block, is a stack of six decoders(Nx = 6), containing two multi-head attention layers. The first multi-head attention layer uses a masked operation. The K and V matrices of the second multi-head attention layer are computed using the encoder's encoded information matrix C, while Q is computed using the output of the previous decoder block. It is the same as the encoder's multi-head attention computation, but with the addition of a mask code, which is a mask that masks certain values so that they do not have an effect when the parameters are updated. There are two types of masks involved in the Transformer model [37], namely, the padding mask and the sequence mask.

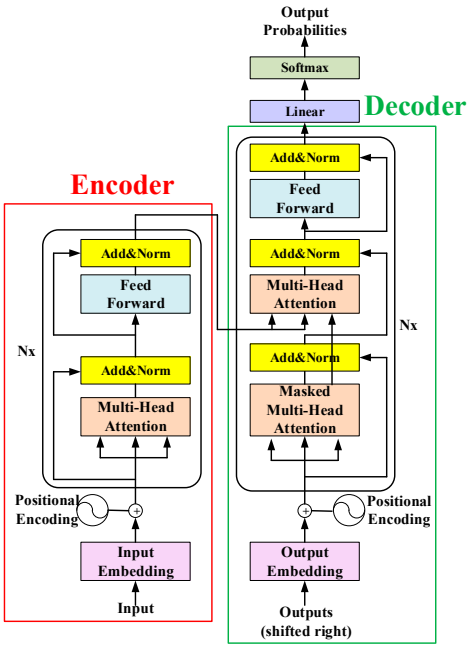


**Figure 3.** Transformer algorithm architecture.

In the final output of the algorithm, it first undergoes a linear transformation (the linear transformation layer is a simple fully connected neural network that projects the vector produced by the decoding component into a much larger vector called the log-odds vector), and then softmax obtains the probability distribution of the output.

### 3.1.3. BiLSTM Algorithm Key Technology

The LSTM consists of forget gates, input gates, and output gates to protect and control the state of the neurons, and with the addition of the cell state, the structure of the long and short-term memory network is shown in Figure 4.

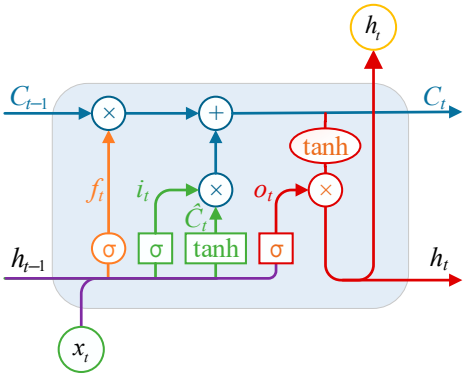


**Figure 4.** LSTM algorithm structure diagram.

The forget gate determines which information is forgotten in this cell state, and the forget gate of LSTM is defined as shown below:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{11}$$

where $\sigma$ denotes the activation function, $h_{t-1}$ is the hidden layer output at moment $t-1$, $x_t$ is the input data at moment $t$, $b_f$ is the deviation at moment $t$, and $W_f$ is the weight matrix of the forgetting gate. The activation function is usually a sigmoid function, which is defined as shown below:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{12}$$

The sigmoid function is used as an activation function to map the variables between 0 and 1. The outputs of $h_{t-1}$ and $x_t$ in the forgetting gate are determined by the weights $W_f$. When $f_t$ is 1, it indicates the complete retention of $C_{t-1}$, and an output of 0 indicates complete forgetting.

The input gate will determine the input of new information into the cell state. The LSTM input gate is defined as shown below:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{13}$$

$$\hat{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{14}$$

where $W_i$ denotes the weight of the input gate, $b_i$ is the deviation of the input gate, $W_c$ denotes the weight of the new cell, and $b_c$ denotes the deviation of the new cell. The importance of the input data is selected by the weights $W_i$, $i_t$ denotes the proportion of the input information, while $h_{t-1}$ and $x_t$ are passed through tanh to create a new candidate cell $\hat{C}_t$, which is subsequently added to the cell state.

The new cell state, which is a combination of the old cell state retained by the forget gate and the new candidate cell added by the input gate, is defined as shown below:

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \tag{15}$$

The output cell determines which part of the updated cell state will be output, and the LSTM output gate is defined as shown below:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{16}$$

$$h_t = o_t * \tanh(C_t) \tag{17}$$

where $W_o$ is the weight of the output gate and $b_o$ is the deviation of the output gate. The cell state $C_t$ decides what information to output through the tanh activation function, which is then multiplied by the proportion of information output from the output gate $o_t$ to obtain the final hidden layer output $h_t$.

BiLSTM, on the other hand, adds a reaction process to the original unidirectional LSTM to capture the distribution of features from the future to the present.

### *3.2. Architecture of RUL Prediction Algorithm Based on KAN–Transformer–BiLSTM*

In the preparation of the dataset, FD001 and FD003 from NASA's C-MAPSS dataset were used as the data source to support the characterization parameters including aircraft number, timestamp, flight cycle, engine bearing temperature, engine pressure ratio, engine rotational rate, bearing tile temperature, operating fluid temperature, operating fluid pressure, operating fluid flow rate, high pressure-bearing temperature, fuel flow rate, rotor speed, outlet pressure ratio, fuel temperature, fuel pressure, pressurizer outlet temperature, pressurizer outlet pressure, internal bearing temperature, pressurizer flow rate, pressurizer turbine temperature, thrust, thrust modification, engine load, and engine vibration, totaling 25 feature parameters used as input to the full feature dataset. Among them, three features, namely, aircraft number, timestamp, and flight cycle, are not directly related to RUL, so only the remaining 22 feature parameters are selected for the reconstruction of the dataset.

In FD001, the samples of the training parameters contain 20,631, while the samples of the test parameters contain 13,096, so the division of the total samples between the two is carried out according to the ratio of 0.388, and the training dataset is fused with the test set to divide the training set, the validation set, and the test set of the new samples according to the ratio of 0.7, 0.2, and 0.1, which serves as the input to the fusion algorithm.

In the execution process of the algorithm, the first part of the Transformer algorithm is used. In order to satisfy the number of features and samples in the training parameters, the embedding dimension of the Transformer's algorithm is set to be 32, the number of hidden layers is set to be 32, the number of heads of the multi-head attention mechanism is set to be 8, the deactivation rate of the dropout layer is set to be 0.2, the number of encoder–decoder pairs is set to be 6, the learning rate is set to be 0.0001, and the batch size is 256. The multi-step prediction algorithm is used in the algorithm, which predicts the remaining lifetime prediction value of a subsequent time window through ten steps, and the number of iterations of the algorithm is 50. In the second part of the algorithm, the KAN algorithm is used, and for the number of grids in the KAN algorithm, we used an initial value of 200, which takes advantage of the dual advantages of the MLP algorithm and B-spline. The MLP algorithm and B-spline are used to construct the relationship between the 22-dimensional aero-engine characteristic parameters and the number of rounds of remaining useful life, where the MLP algorithm and the B-spline solve the relationship between the long-term dependence and the short-term dependence, respectively. The use conditions of the two correspond to the following cases, respectively. In the feature dataset,

there are a small number of features that remain unchanged or have a weak trend of change throughout the engine operating cycle, and this belongs to the short-term dependence because of the weak conditions of influence in terms of the long-term process. In contrast, the remaining feature parameters have a strong trend of change with the progression of the operating time of the cycle, and this feature is called the long-term dependence. The KAN algorithm is mainly used to capture the relationship between these two types of feature parameters to establish a high-dimensional mapping between them and the desired output RUL values by taking advantage of the MLP and B-spline targeting, respectively. The role of BiLSTM, on the other hand, employs the advantage of bilateral channels for bidirectional feature capture from past to current and future to current, which provides a detailed construction and mapping relationship to further enhance the mapping between features and predicted values.

The chosen hyperparameters are aligned with common practices for Transformer models in time-series forecasting. They ensure that the model is both powerful enough to learn complex temporal dependencies and efficient enough to avoid overfitting and excessive computational costs. The specific values were selected based on prior research in time-series tasks as well as experimentation to achieve the best balance between model accuracy and training efficiency.

To improve methodological transparency, the overall workflow of the proposed Transformer–KAN–BiLSTM framework is summarized in Figure 5. The process begins with data preprocessing, including sensor normalization, the removal of constant channels, and statistical smoothing to handle noisy readings. The preprocessed data are then partitioned into training (70%), validation (20%), and test (10%) subsets. Subsequently, the Transformer module performs feature embedding and long-range dependency extraction; the KAN module constructs adaptive spline-based mappings to enhance nonlinear feature learning; and the BiLSTM module captures bidirectional temporal correlations. Finally, the integrated model undergoes performance evaluation using RMSE, MAE, MAPE, and $R^2$ metrics, which is followed by statistical significance testing to confirm robustness. This structured workflow ensures consistency and reproducibility across all datasets.

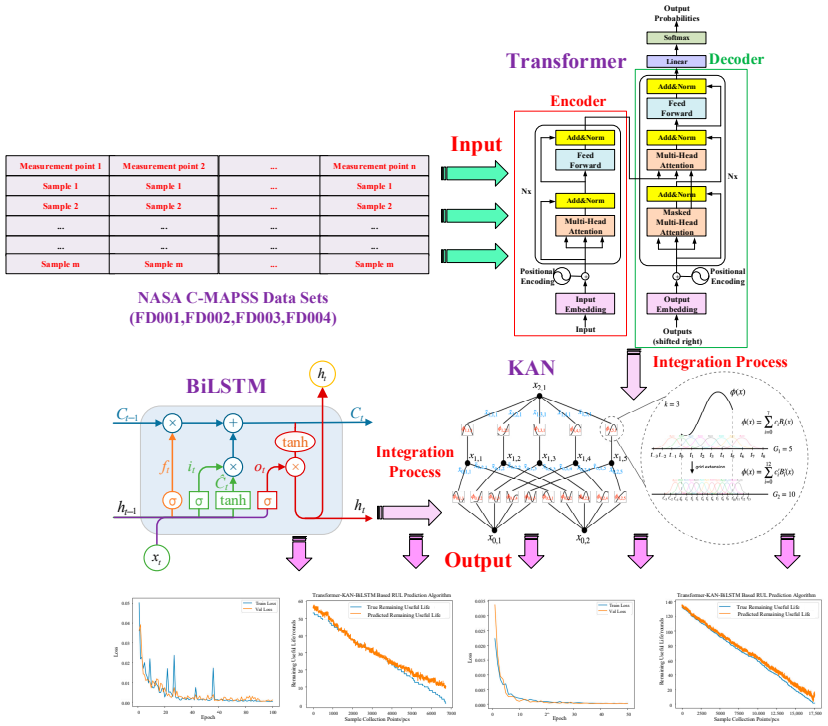


**Figure 5.** Based on Transformer + KAN + BiLSTM algorithm architecture.

## 4. Validation of RUL Prediction Algorithm Results

### *4.1. Overview of KAN Algorithm Interpretability and Generalization*

The KAN algorithm's interpretability mainly consists of the following parts: first, there is its sparsification processing logic. The KAN algorithm adopts a learnable activation function to replace the original linear weights based on the MLP. It uses the L1 paradigm to define the expression of its activation function and improves it by entropy regularization.

The L1 paradigm of the activation function $\phi$ is defined over the average amplitude of its $N_p$ inputs, i.e., $|\phi|_1 \equiv \frac{1}{N_p}\sum_{s=1}^{N_p}\left|\phi\left(x^{(s)}\right)\right|$. For the KAN layer $\phi$ with 22 inputs and 1 output, the L1 paradigm of $\phi$ is defined as the sum of the L1 paradigms of all the activation functions, i.e., $\Phi\Big|_1 \equiv \sum_{i=1}^{n_{\text{in}}}\sum_{j=1}^{n_{\text{out}}}\left|\phi_{i,j}\right|_1$, and we define the entropy of $\phi$ to be $S(\mathbf{\Phi}) \equiv -\sum_{i=1}^{n_{\text{in}}}\sum_{j=1}^{n_{\text{out}}}\frac{|\phi_{i,j}|_1}{|\mathbf{\Phi}|_1}\log\left(\frac{|\phi_{i,j}|_1}{|\mathbf{\Phi}|_1}\right)$. The total training objective $T_{total}$ is then L1 and the entropy regularization of $T_{pred}$ with all KAN layers $T_{\text{total}} = T_{\text{pred}} + \lambda\left(\mu_1\sum_{l=0}^{L-1}|\mathbf{\Phi}_l|_1 + \mu_2\sum_{l=0}^{L-1}S(\mathbf{\Phi}_l)\right)$: where μ1, μ2 are relative sizes usually set to μ1 = μ2 = 1, and λ controls the overall regularization amplitude.

The second part is its visual expression. For a KAN to be visualized in such a way that the magnitude can reveal the expression, the transparency of the activation function $\phi_{l,i,j}$ is set as a proportion of $\tanh\left(\beta A_{l,i,j}\right)$. When insignificant expressions appear, they are ignored, and the significant function can be revealed. In the pruning operation, on the other hand, after training with a sparsification penalty, it is generally necessary to also prune the network to a smaller sub-network. The KAN is sparsified at the node level, and for each node, its incoming and outgoing scores are defined as $I_{l,i} = \max_k\left(|\phi_{l-1,k,i}|_1\right)$, $\quad O_{l,i} = \max_j\left(\left|\phi_{l+1,j,i}\right|_1\right)$. Considering whether a node is important or not, the pruning operation takes effect if both the incoming and outgoing scores are greater than a threshold hyperparameter bar $\theta = 10^2$. The final step is symbolization optimization, using $fix_symbolic(l,i,j,f)$ to set the $(l,i,j)$ activation to $f$, obtaining the pre-activation x and post-activation from the sample, and fitting the affine parameter $(a,b,c,d)$ to make $y \approx cf(ax+b)+d$.

Generalization ability and robustness analysis mainly include two aspects. In terms of validation results, the datasets of FD002 and FD004 contain the predicted values of aero-engine characteristic parameters and their remaining useful life under six operating conditions, and the algorithms have obtained satisfactory results under all six operating conditions. In the validation part, all the data in NASA's C-MAPSS dataset are used to validate the algorithms, and the algorithms are also used to validate the algorithms. In the validation part, all the data in NASA's C-MAPSS dataset, including four datasets, FD001, FD002, FD003, and FD004, are used to validate the final results using the same algorithm, which differs only in the adjustment of hyperparameters of the algorithm, while the algorithmic structure, the design concepts, and the technological routes are all the same. The final results are shown in Figures 6–9, which correspond to the R prediction results of FD001, FD002, FD003, and FD004, respectively.

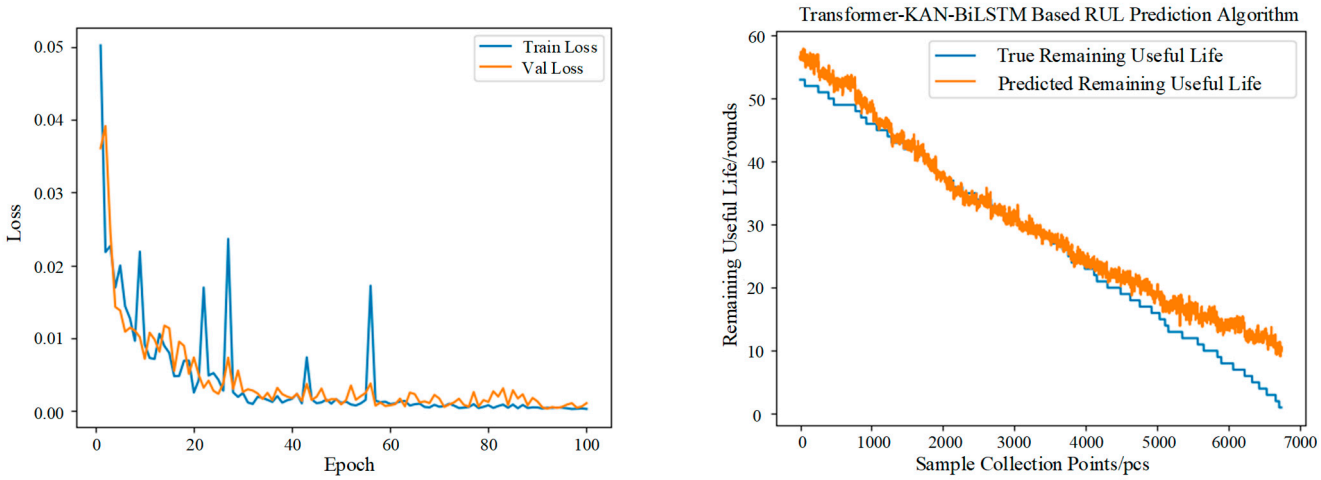


**Figure 6.** RUL prediction results for the FD001 dataset.

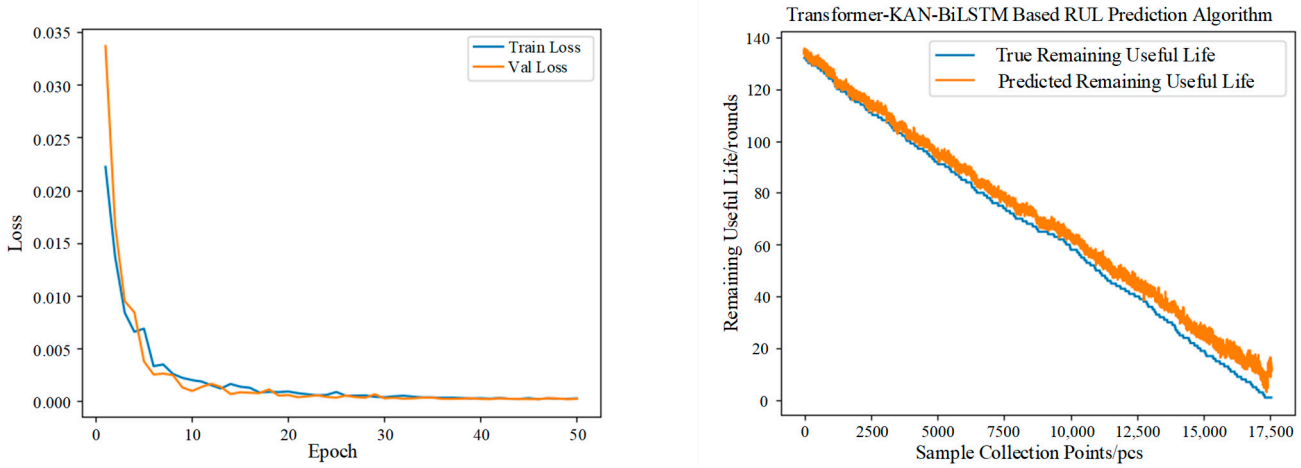


**Figure 7.** RUL prediction results for the FD002 dataset.

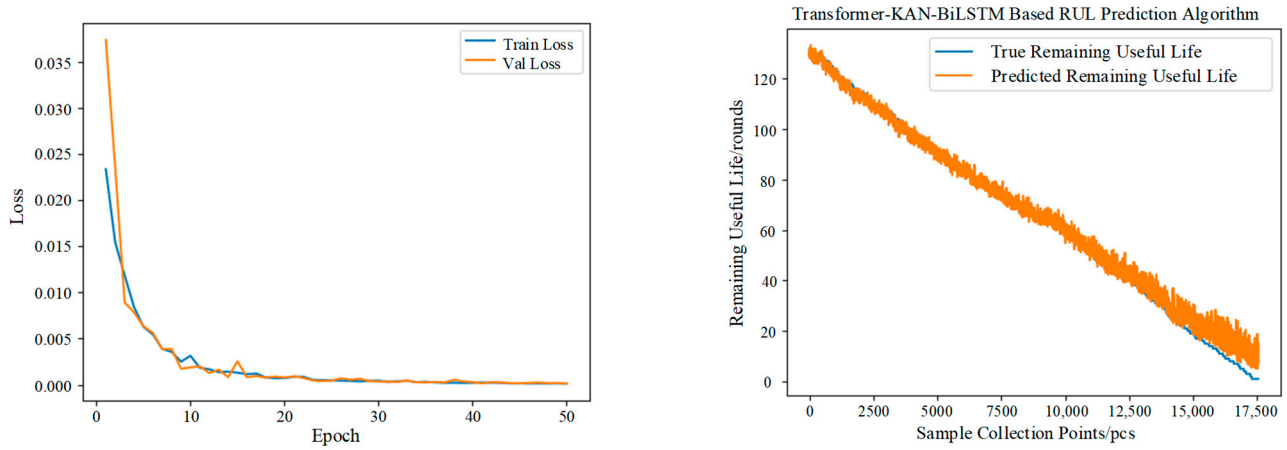


**Figure 8.** RUL prediction results for the FD003 dataset.

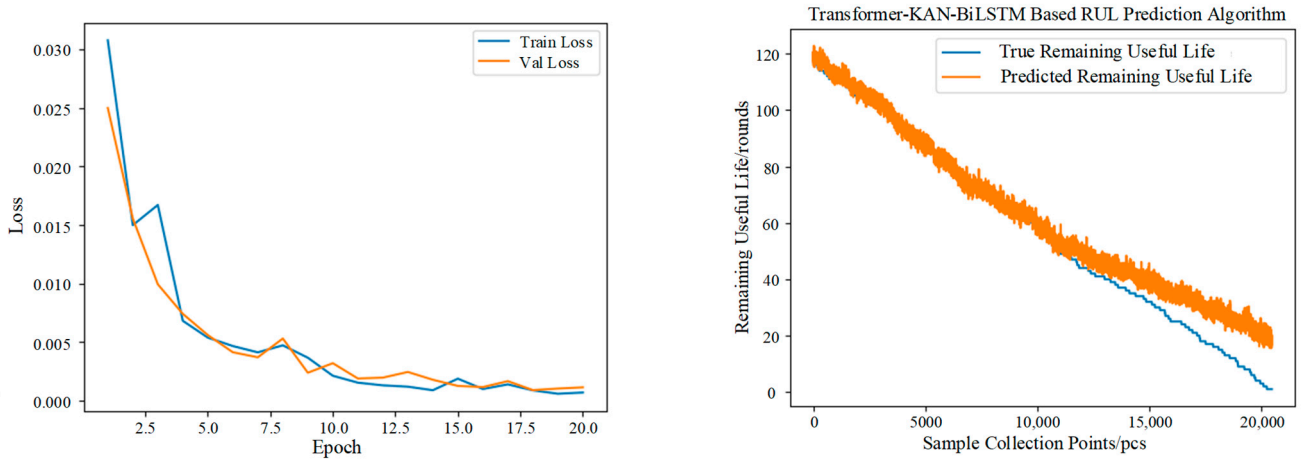


**Figure 9.** RUL prediction results for the FD004 dataset.

The results shown in Figures 6–9 show that, finally, the test set results obtained after the algorithm prediction and the actual RUL value meet the needs of engineering and innovative concepts, indicating the innovativeness, completeness and accuracy of the research algorithms studied in this paper. Moreover, since both FD002 and FD004 are RUL variations due to the degradation of aero-engine characteristic parameters under multiple operating conditions, the parameter samples are highly heterogeneous, and thus the results are somewhat more biased compared to FD001 and FD003. However, in order to achieve better results and improve the significance, the number of grids in the KAN algorithm is set to a value between 40 and 50, the hidden neurons in the Transformer architecture are improved to 64 and 128, respectively, and the number of heads in the multi-head attention mechanism is also set to 8 and 16, respectively.

### *4.2. Comparative Results Analysis and Validation*

To evaluate the performance of the proposed Transformer–KAN–BiLSTM model for RUL prediction, we used the following standard metrics: Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE) and the Coefficient of Determination ($R^2$). The combination of RMSE, MAPE, MAE and $R^2$ provides a well-rounded evaluation of model performance in RUL prediction. Each metric highlights a different aspect of model accuracy. The RMSE emphasizes the importance of avoiding large errors, which is critical for safety and reliability in engineering applications. The MAPE provides a normalized measure that is easy to interpret and useful when comparing performance across different datasets or operational conditions. The MAE gives a straightforward, average measure of error, which is useful for understanding the typical error magnitude in practical terms. $R^2$ is included to quantify the proportion of variance in the true RUL that is explained by our model. A high $R^2$ value indicates that the model successfully captures the underlying degradation trend of the engines.

Table 4 lists the evaluation index values of RMSE, R2, MAPE, and MAE in FD001, FD002, FD003, and FD004, respectively. Table 5 lists the evaluation indices of other public algorithms and experimental algorithms of the control group under the same conditions during the study, so that it can be analyzed and determined that the algorithms investigated herein represent RMSE and MAPE.

**Table 4.** Analysis of the results of the algorithmic evaluation indicators.

| Dataset | R2 | RMSE | MAE | MAPE |
|---|---|---|---|---|
| FD001 | 0.9806 | 3.6784 | 3.3256 | 0.2023 |
| FD002 | 0.9823 | 4.9498 | 4.5000 | 0.3284 |
| FD003 | 0.9896 | 3.7988 | 2.7771 | 0.3015 |
| FD004 | 0.9489 | 7.5202 | 5.4508 | 0.5982 |

**Table 5.** Comparative test results and evaluation index analysis.

| Algorithms | RMSE | MAPE | *p*-Value (vs. Transformer–KAN) | Inference Time (ms/Sample) | FLOPs (G) |
|---|---|---|---|---|---|
| LSTM | 24.236 | 2.1234 | <0.001 | 1.42 | 2.4 |
| Deep-Layer LSTM | 17.236 | 1.4623 | <0.001 | 1.65 | 2.9 |
| LSTM–KAN | 15.159 | 1.2569 | <0.001 | 1.78 | 3.1 |
| BiLSTM | 22.658 | 1.9754 | <0.001 | 1.63 | 3.0 |
| BiLSTM–KAN | 15.032 | 1.2236 | <0.001 | 1.92 | 3.3 |
| GRU | 25.145 | 2.3261 | <0.001 | 1.38 | 2.1 |
| GRU–KAN | 17.632 | 1.528 | <0.001 | 1.70 | 2.8 |
| TCN | 21.364 | 1.9023 | <0.001 | 1.55 | 2.5 |
| TCN–KAN | 14.326 | 1.1986 | <0.001 | 1.83 | 3.2 |
| Transformer–KAN (Baseline) | 7.634 | 0.5361 | - | 2.21 | 3.8 |
| Transformer | 12.364 | 0.9963 | 0.014 | 2.04 | 3.6 |
| Transformer–KAN—BiLSTM | 3.6784 | 0.2023 | <0.001 | 2.68 | 4.5 |

The FD001 dataset was used as the data source input for the comparison algorithm, and control variables were used to analyze the comparison results under the same conditions.

To assess whether the observed improvements were statistically meaningful, paired *t*-tests were performed on RMSE and MAPE values over ten independent runs, comparing each competing method with the Transformer–KAN baseline. The results (Table 5) show $p < 0.05$ for all comparisons, confirming that the proposed Transformer–KAN–BiLSTM model yields statistically significant performance gains.

To evaluate computational efficiency in practical deployment, Table 5 also reports the inference time and theoretical FLOPs for each model. The proposed Transformer–KAN–BiLSTM requires approximately 2.68 ms per sample compared with 2.21 ms for the Transformer–KAN baseline and 1.4–2.0 ms for other models. Although the hybrid model shows a modest 20% increase in inference cost, it achieves a more than 50% lower RMSE and 60% lower MAPE, indicating a favorable balance between prediction accuracy and computational complexity. This demonstrates that the proposed architecture maintains real-time feasibility while significantly improving reliability.

To concretely demonstrate the interpretability of the KAN component, Figure 10 presents the learned univariate spline transformations and their first-order derivatives for representative engine features, including the turbine inlet temperature, compressor outlet pressure, and fuel flow rate. Each subplot visualizes the spline-based activation $g(x)$ learned by the KAN and its derivative $g'(x)$, revealing how the model internally encodes nonlinear feature–RUL relationships. The vertical dashed lines indicate spline knot positions, allowing the visualization of monotonic or oscillatory patterns corresponding to distinct degradation behaviors—such as steady thermal wear versus cyclic operational effects.

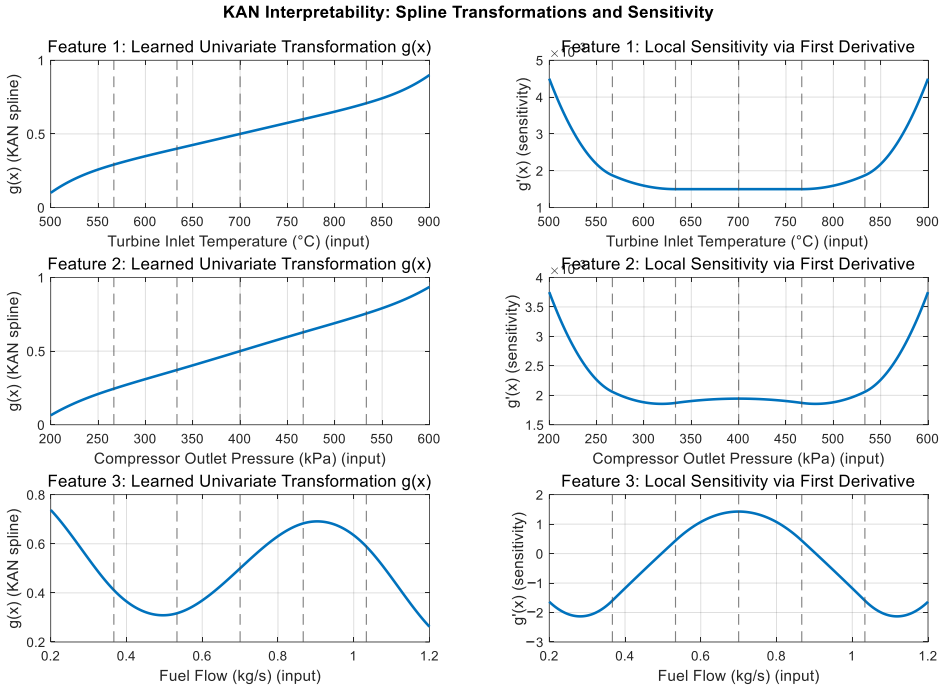


**Figure 10.** Visualization of KAN interpretability.

These results demonstrate that the KAN not only improves predictive accuracy but also provides transparent, feature-level interpretability, where physical variables can be directly mapped to their functional influence on RUL estimation. This is consistent with recent studies emphasizing visualization as an essential aspect of model interpretability.

## 5. Discussion

The results of this study demonstrate that the proposed Transformer–KAN–BiLSTM fusion model significantly outperforms existing methods in predicting the remaining useful life (RUL) of aero-engines across multiple datasets (FD001–FD004). The model achieves notably lower RMSE and MAPE values compared to traditional and hybrid baselines, indicating superior accuracy and robustness. These improvements can be attributed to the complementary strengths of the three integrated algorithms: the Transformer's ability to capture long-term dependencies and parallelize computation, KAN's high-dimensional mapping and interpretability, and BiLSTM's bidirectional feature extraction for short-term trends.

From the perspective of previous studies, many existing approaches—such as standalone LSTM, GRU, TCN, or even their hybrid forms—often struggle with data heterogeneity, overfitting, and limited generalization. Our model addresses these issues through structured multimodal fusion, which not only improves prediction accuracy but also enhances model interpretability via KAN's symbolic and visual capabilities.

The implications of this work extend beyond aero-engine prognostics. The proposed framework offers a generalizable blueprint for multimodal time-series prediction in other high-stakes domains such as wind turbine health monitoring, nuclear plant safety systems, and medical device maintenance, where interpretability and accuracy are equally critical. Furthermore, the integration of symbolic AI elements (via KAN) with deep learning architectures represents a step toward more transparent and trustworthy AI systems in engineering applications.

The proposed Transformer–KAN–BiLSTM framework effectively mitigates data heterogeneity through its hierarchical and complementary structure. The Transformer module

employs a multi-head attention mechanism that automatically re-weights heterogeneous sensor features according to their contextual relevance, thereby reducing the inter-sensor imbalance caused by varying noise levels or operating conditions. The KAN component, equipped with adaptive B-spline activations, locally adjusts its response to each feature's statistical scale and nonlinearity, providing flexible feature-specific mappings. Finally, the BiLSTM captures temporal continuity and inter-feature correlations, ensuring consistent degradation representation even when different parameters evolve asynchronously. Collectively, these components perform a form of implicit domain adaptation, allowing the model to maintain robustness and accuracy across multiple C-MAPSS subsets with differing operating regimes and fault modes.

The performance of the proposed Transformer–KAN–BiLSTM model is critically evaluated against a range of recent and representative studies on the C-MAPSS benchmark, as summarized in Table 5. This comparative analysis highlights the distinct advantages of our multimodal fusion strategy.

While deep learning models like Deep-Layer LSTM [4] have demonstrated strong capabilities in capturing temporal patterns, they often struggle with the long-term dependencies and complex, nonlinear feature interactions present in full-life cycle engine data. Our model addresses this by integrating the Transformer, whose self-attention mechanism is inherently more suited for modeling long-range contextual relationships, leading to a significant reduction in RMSE (e.g., from 12.56 for Deep-Layer LSTM on FD001 to our 3.6784).

Recent hybrid models have attempted to overcome these limitations. For instance, Wang et al. [15] combined a 1D-CNN with a BiLSTM-AM, effectively capturing local spatial features and short-term temporal dependencies. However, the absence of a dedicated component for long-term dependency modeling can limit its performance on sequences with gradual degradation trends. In contrast, our framework explicitly incorporates the Transformer for this purpose. Similarly, Guo et al. [17] proposed a multi-scale Hourglass-Transformer, which excels at multi-resolution feature fusion. Our approach complements this direction by introducing the KAN as a superior alternative to MLPs for high-dimensional mapping, offering not only enhanced accuracy but also a pathway to interpretability, which their model lacks.

Furthermore, when compared to other models that integrate modern architectures, our fusion shows clear benefits. The Transformer–KAN variant in our ablation study already outperforms standalone Transformers and other hybrids like TCN–KAN, underscoring the unique value of replacing static MLP layers with adaptive KANs. The full Transformer–KAN–BiLSTM model then achieves the best overall performance by further incorporating bidirectional fine-grained temporal analysis, which is a feature that is missing in broader architectures like MSBLS [20].

The proposed Transformer–KAN–BiLSTM model demonstrates excellent performance, achieving notably low RMSE and MAPE values on the C-MAPSS dataset. However, it is important to consider the potential risks of overfitting, especially given the complexity of the model and its outstanding performance on the training data. To mitigate overfitting, we employed several strategies during the training process. We applied a dropout rate of 0.2 in the model to regularize the network and prevent it from relying too heavily on specific neurons, which could lead to overfitting. To further combat overfitting, we used early stopping during training, halting training once the validation loss started to increase, indicating that the model was beginning to overfit. We used cross-validation to ensure that the model's performance was not dependent on a particular training subset, improving its generalizability. The performance was evaluated on multiple subsets (such as FD001, FD002, FD003, and FD004) to validate that the model could generalize across different

datasets. Despite the potential for overfitting, the model performed well on the test set, which was not used during training. This indicates that the model generalizes effectively to new, unseen data. The performance on the test set (which included data from FD001, FD002, FD003, and FD004) was consistently good, demonstrating that the model's predictions were not merely memorizing the training data.

The primary limitation of this study is the use of the C-MAPSS dataset, which, while widely used and valuable, is synthetic and does not fully capture the complexities of real-world engine data. As the data are generated through simulation, it lacks the noise, sensor imperfections, and unpredictable operating conditions that are typically found in real-world applications. Additionally, the dataset only includes relatively simple fault modes, which may not represent the full range of degradation scenarios that can occur in operational environments. Furthermore, the dataset's controlled nature may lead to the risk of overfitting, which limits the generalizability of the model to real-world cases. In addition, it does not provide data on components such as the Accessory Gearbox, Engine Mounting System, and Nacelle. These are essential systems in an aero-engine, and their failure modes could significantly impact the RUL and overall functionality of the engine.

Despite these limitations, the C-MAPSS dataset provides a solid benchmark for model evaluation. However, to ensure the applicability and robustness of our proposed Transformer–KAN–BiLSTM model in practical scenarios, future work should focus on testing and validating the model with real-world data, which would involve more complex fault interactions, diverse operating conditions, and sensor noise.

It should be noted that this study validates the proposed Transformer–KAN–BiLSTM model only on the C-MAPSS dataset. Although the dataset is highly reliable and widely used in the field, cross-dataset transfer testing and domain adaptation analyses have not been performed in this work. Future research will extend this study to real-world datasets and domain adaptation tasks to further verify the generalizability of the model.

## 6. Conclusions

In this paper, the fusion algorithm based on Transformer + KAN + BiLSTM is mainly utilized for the study of a prediction algorithm for RUL across four subsets in the C-MAPSS dataset: FD001, FD002, FD003, and FD004. The main contribution points and conclusions are as follows.

First, a Transformer-based algorithm was used to first input the C-MAPSS dataset a priori to capture the high-intensity dependencies in it and ensure the initial dependency model construction by constructing a strong mapping relationship between the feature dataset, the RUL, and the corresponding multisemantic analysis.

Second, the strong mapping relationship construction process based on the KAN algorithm was adopted. By combining the fitting and mapping relationship construction of the MLP and the B-spline in the long-term dependence and short-term dependence, respectively, the feature parameters in FD001, FD002, FD003, and FD004 were artificially classified into the mutation cases of small change over time, stable unchanged, and large fluctuation of change. The strong mapping model was constructed by utilizing the respective advantages of MLP and B-spline to predict the characteristic parameters and the final RUL values.

Finally, the two-channel concept of the BiLSTM algorithm was finally utilized to capture the completeness of the dataset with bidirectional features, extracting and further enhancing the capability of the previous strong mapping model based on the combination of Transformer and KAN, resulting in ideal predictions of the final RUL, with the final RMSE and MAPE values being in the ideal conditions.

While the model shows promising results, several avenues for future research can be explored:

(1) One of the next critical steps is to apply the Transformer–KAN–BiLSTM model to real-world aero-engine data. The C-MAPSS dataset is synthetic, and real-world data will introduce additional challenges such as noise, sensor failures, and unstructured degradation patterns. Testing the model on operational engine data will help evaluate its robustness in real-world scenarios and improve its practical applicability.

(2) The model could benefit from enhanced domain adaptation techniques, particularly in cases where the model is trained on one subset of data and applied to a different set (e.g., from FD001 to FD004). Future work could explore methods like transfer learning and fine-tuning to improve the model's ability to adapt to new datasets with diverse fault modes and operational conditions.

(3) While the proposed model performs well in terms of prediction accuracy, its computational cost could be reduced to make it more feasible for real-time deployment in predictive maintenance systems. Techniques like model pruning, quantization, or developing lightweight versions of the model could make the model more efficient and suitable for deployment on edge devices with limited computational resources.

**Author Contributions:** Conceptualization, K.X.; methodology, Q.Z.; software, K.X.; validation, K.X.; formal analysis, K.X.; investigation, K.X.; resources, K.X.; data curation, K.X.; writing—original draft preparation, K.X.; writing—review and editing, Q.Z.; visualization, K.X.; supervision, Y.G.; project administration, Y.G.; funding acquisition, Y.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This study was funded by the National Science and Technology Major Program (J2019-V-0003-0094).

**Data Availability Statement:** The data presented in this study are openly available at NASA's data repository, CMAPSS Jet Engine Simulated Data, at https://data.nasa.gov/dataset/cmapss-jet-engine-simulated-data (accessed on 20 October 2025).

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

| | |
|---|---|
| RUL | Remaining Useful Life |
| KAN | Kolmogorov-Arnold Network |
| BiLSTM | Bidirectional Long Short-Term Memory |
| MLP | Multi-Layer Perceptron |
| CNN | Convolutional Neural Network |
| RMSE | Root Mean Square Error |
| MAPE | Mean Absolute Percentage Error |
| MAE | Mean Absolute Error |
| C-MAPSS | Commercial Modular Aero-Propulsion System Simulation |

## References

1. Shen, Y.; Khorasani, K. Hybrid multi-mode machine learning-based fault diagnosis strategies with application to aircraft gas turbine engines. *Neural Netw.* **2020**, *130*, 126–142. [CrossRef]
2. Huang, Y.; Tao, J.; Sun, G.; Wu, T.; Yu, L.; Zhao, X. A novel digital twin approach based on deep multimodal information fusion for aero-engine fault diagnosis. *Energy* **2023**, *270*, 126894. [CrossRef]
3. Jimenez, J.J.M.; Schwartz, S.; Vingerhoeds, R.; Grabot, B.; Salaün, M. Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics. *J. Manuf. Syst.* **2020**, *56*, 539–557. [CrossRef]
4. Ellefsen, A.L.; Bjørlykhaug, E.; Æsøy, V.; Ushakov, S.; Zhang, H. Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. *Reliab. Eng. Syst. Saf.* **2019**, *183*, 240–251. [CrossRef]

5. Deng, K.; Zhang, X.; Cheng, Y.; Zheng, Z.; Jiang, F.; Liu, W.; Peng, J. A remaining useful life prediction method with long-short term feature processing for aircraft engines. *Appl. Soft Comput.* **2020**, *93*, 106344. [CrossRef]
6. Li, X.; Ding, Q.; Sun, J.Q. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliab. Eng. Syst. Saf.* **2018**, *172*, 1–11. [CrossRef]
7. Liu, L.; Song, X.; Zhou, Z. Aircraft engine remaining useful life estimation via a double attention-based data-driven architecture. *Reliab. Eng. Syst. Saf.* **2022**, *221*, 108330. [CrossRef]
8. Cheng, Y.; Zeng, J.; Wang, Z.; Song, D. A health state-related ensemble deep learning method for aircraft engine remaining useful life prediction. *Appl. Soft Comput.* **2023**, *135*, 110041. [CrossRef]
9. Begni, A.; Dini, P.; Saponara, S. Design and test of an lstm-based algorithm for li-ion batteries remaining useful life estimation. In Proceedings of the International Conference on Applications in Electronics Pervading Industry, Environment and Society, Genova, Italy, 26–27 September 2022; Springer Nature: Cham, Switzerland, 2022; pp. 373–379.
10. Djeziri, M.A.; Benmoussa, S.; Sanchez, R. Hybrid method for remaining useful life prediction in wind turbine systems. *Renew. Energy* **2018**, *116*, 173–187. [CrossRef]
11. Abid, K.; Sayed-Mouchaweh, M.; Cornez, L. Adaptive data-driven approach for the remaining useful life estimation when few historical degradation sequences are available. In Proceedings of the 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 14–17 December 2020; pp. 1145–1152.
12. Cubillo, A.; Perinpanayagam, S.; Esperon-Miguez, M. A review of physics-based models in prognostics: Application to gears and bearings of rotating machinery. *Adv. Mech. Eng.* **2016**, *8*, 1687814016664660. [CrossRef]
13. Lei, Y.; Li, N.; Guo, L.; Li, N.; Yan, T.; Lin, J. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mech. Syst. Signal Process.* **2018**, *104*, 799–834. [CrossRef]
14. Wu, J.; Su, Y.; Cheng, Y.; Shao, X.; Deng, C.; Liu, C. Multi-sensor information fusion for remaining useful life prediction of machining tools by adaptive network based fuzzy inference system. *Appl. Soft Comput.* **2018**, *68*, 13–23. [CrossRef]
15. Wang, L.; Chen, Y.; Zhao, X.; Xiang, J. Predictive maintenance scheduling for aircraft engines based on remaining useful life prediction. *IEEE Internet Things J.* **2024**, *11*, 23020–23031. [CrossRef]
16. Xu, T.; Han, G.; Zhu, H.; Lin, C.; Peng, J. Multiscale BLS-based lightweight prediction model for remaining useful life of aero-engine. *IEEE Trans. Reliab.* **2024**, *73*, 1757–1767. [CrossRef]
17. Guo, J.; Lei, S.; Du, B. MHT: A multiscale hourglass-transformer for remaining useful life prediction of aircraft engine. *Eng. Appl. Artif. Intell.* **2024**, *128*, 107519. [CrossRef]
18. Sharma, R.K. Framework Based on Machine Learning Approach for Prediction of the Remaining Useful Life: A Case Study of an Aviation Engine. *J. Fail. Anal. Prev.* **2024**, *24*, 1333–1350. [CrossRef]
19. Arunan, A.; Qin, Y.; Li, X.; Yuen, C. A change point detection integrated remaining useful life estimation model under variable operating conditions. *Control Eng. Pract.* **2024**, *144*, 105840. [CrossRef]
20. Furqon, M.; Pratama, M.; Liu, L.; Habibullah, H.; Dogancay, K. Mixup domain adaptations for dynamic remaining useful life predictions. *Knowl.-Based Syst.* **2024**, *295*, 111783. [CrossRef]
21. Maulana, F.; Starr, A.; Ompusunggu, A.P. Explainable data-driven method combined with bayesian filtering for remaining useful lifetime prediction of aircraft engines using nasa cmapss datasets. *Machines* **2023**, *11*, 163. [CrossRef]
22. Somvanshi, S.; Javed, S.A.; Islam, M.M.; Pandit, D.; Das, S. A survey on kolmogorov-arnold network. *ACM Comput. Surv.* **2024**, *58*, 1–35. [CrossRef]
23. Sulaiman, M.H.; Mustaffa, Z.; Saealal, M.S.; Saari, M.M.; Ahmad, A.Z. Utilizing the Kolmogorov-Arnold Networks for chiller energy consumption prediction in commercial building. *J. Build. Eng.* **2024**, *96*, 110475. [CrossRef]
24. Sulaiman, M.H.; Mustaffa, Z.; Mohamed, A.I.; Samsudin, A.S.; Rashid, M.I.M. Battery state of charge estimation for electric vehicle using Kolmogorov-Arnold networks. *Energy* **2024**, *311*, 133417. [CrossRef]
25. Wang, S.; Shi, J.; Yang, W.; Yin, Q. High and low frequency wind power prediction based on Transformer and BiGRU-Attention. *Energy* **2024**, *288*, 129753. [CrossRef]
26. Feng, Z.; Zhang, J.; Jiang, H.; Yao, X.; Qian, Y.; Zhang, H. Energy consumption prediction strategy for electric vehicle based on LSTM-transformer framework. *Energy* **2024**, *302*, 131780. [CrossRef]
27. Al-Majali, M.R.; Zhang, M.; Al-Majali, Y.T.; Trembly, J.P. Impact of raw material on thermo-physical properties of carbon foam. *Can. J. Chem. Eng.* **2025**, *103*, 1309–1318. [CrossRef]
28. Guo, J.; Liu, M.; Luo, P.; Chen, X.; Yu, H.; Wei, X. Attention-based BILSTM for the degradation trend prediction of lithium battery. *Energy Rep.* **2023**, *9*, 655–664. [CrossRef]
29. Zhao, J.; Zhang, M.; Wang, C.; Yu, W.; Zhu, Y.; Zhu, P. First-Principles Study of CO, $NH_3$, HCN, CNCl, and $Cl_2$ Gas Adsorption Behaviors of Metal and Cyclic C–Metal B-and N-Site-Doped h-BNs. *Electron. Mater. Lett.* **2025**, *21*, 268–288. [CrossRef]
30. Ferreira, C.; Gonçalves, G. Remaining Useful Life prediction and challenges: A literature review on the use of Machine Learning Methods. *J. Manuf. Syst.* **2022**, *63*, 550–562. [CrossRef]

31. Vollert, S.; Theissler, A. Challenges of machine learning-based RUL prognosis: A review on NASA's C-MAPSS data set. In Proceedings of the 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ET-FA), Västerås, Sweden, 7–10 September 2021; pp. 1–8.
32. Kim, G.; Choi, J.G.; Lim, S. Using transformer and a reweighting technique to develop a remaining useful life estimation method for turbofan engines. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108475. [CrossRef]
33. DeCastro, J.; Litt, J.; Frederick, D. A modular aero-propulsion system simulation of a large commercial aircraft engine. In Proceedings of the 44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, Hartford, CT, USA, 21–23 July 2008; p. 4579.
34. Liu, H.; Zhou, S.; Gu, W.; Zhuang, W.; Gao, M.; Chan, C.C.; Zhang, X. Coordinated planning model for multi-regional ammonia industries leveraging hydrogen supply chain and power grid integration: A case study of Shandong. *Appl. Energy* **2025**, *377*, 124456. [CrossRef]
35. He, S.; Ding, L.; Xiong, Z.; Spicer, R.A.; Farnsworth, A.; Valdes, P.J.; Wang, C.; Cai, F.; Wang, H.; Sun, Y.; et al. A distinctive Eocene Asian monsoon and modern biodiversity resulted from the rise of eastern Tibet. *Sci. Bull.* **2022**, *67*, 2245–2258. [CrossRef] [PubMed]
36. Wang, H.; Song, Y.; Yang, H.; Liu, Z. Generalized Koopman Neural Operator for Data-driven Modelling of Electric Railway Pantograph-catenary Systems. *IEEE Trans. Transp. Electrif.* **2025**. *early access*. [CrossRef]
37. Yan, J.; Cheng, Y.; Zhang, F.; Li, M.; Zhou, N.; Jin, B.; Wang, H.; Yang, H.; Zhang, W. Research on multimodal techniques for arc detection in railway systems with limited data. *Struct. Health Monit.* **2025**, 14759217251336797. [CrossRef]


**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.